

# Kocaeli Üniversitesi

## Bilgisayar Mühendisliği Bölümü

### Yazılım Laboratuvarı II

#### Graf Tabanlı Metin Özetleme Projesi

Büyümin Ekşici Ogün Bingöl

[170201014@kocaeli.edu.tr](mailto:170201014@kocaeli.edu.tr) [170201089@kocaeli.edu.tr](mailto:170201089@kocaeli.edu.tr)

*Bu projede verilen bir dokümandaki cümlelerin graf yapısına dönüştürülmesi ve bu graf modelinin görselleştirilmesi istenmektedir. Ardından graf üzerindeki düğümler ile özet oluşturan bir algoritma oluşturulması beklenmektedir.*

**Anahtar Kelimeler:** Doğal Dil İşleme, Graph Analizi, Tokenization, Stemmer, Stopwords

### Projenin Özeti

Yazılım laboratuvarı 2 3.projesi olarak bizden “Graf Tabanlı Metin Özetleme” adındaki bir uygulama yapılması istendi. Bize projeyi tanıtan pdfte açıklanan toplam 5 ana isteri uygulamaya çalıştık. Bunlar Masaüstü Arayüzü Geliştirilmesi ve Graf Yapısının Oluşturulması , Cümleler Arası Anlamsal İlişkinin Kurulması, Cümle Skoru Hesaplama Algoritmasının Geliştirilmesi, Skorlara Göre Metin Özetleme Algoritmasının Geliştirilmesi ve Özetleme Başarısının ROUGE Skoru ile Hesaplanmasıdır.

Biz bu proje için bizden istenenler doğrultusunda programlama dilleri C++, C#, Java veya Python arasından Java’yı, graf yapısını oluşturmak için JUNG Graph kütüphanesini, Web API kullanımı için Jsoup kütüphanesi ve dil işleme için OpenNLP kullanmayı uygun gördük.

Verilen bir dokümandaki cümlelerin graf yapısına dönüştürülmesi ve bu graf modelinin görselleştirilmesi istenmektedir. Ardından graf üzerindeki düğümler ile özet oluşturan bir algoritma oluşturulması beklenmektedir.

## I. GİRİŞ

Projede temel amaç; cümleleri graf yapısına çevirip Cümle Seçerek Özetleme (Extractive Summarization) gerçekleştirmektir. Graf yapısına çevirerek cümlelerin metindeki anlamsal ilişkilerini görselleştirmek ve bu ilişkileri kullanarak önemli cümleleri belirlemek amaçlanmaktadır.

Masaüstü uygulamada ilk olarak doküman yükleme işlemi gerçekleştirilecektir. Ardından yüklenen dokümandaki cümleleri graf yapısı haline getirerek ve bu graf yapısı görselleştirilecektir. Bu grafta her bir cümle bir düğümü temsil edecektir. Cümleler arasındaki anlamsal ilişki kurulmalı, aynı zamanda cümleler skorlanmalıdır. Belirli parametreleri

kullanarak cümle skorunun hesaplama algoritmasını ve cümle skorlarına göre metin özeti çıkarma algoritmaları geliştirilecektir. Özet metni arayüzde sunup sonuç olarak da verilen bir metnin özetini bu yöntem ile çıkarılıp ve gerçek özet ile benzerliğini “ROUGE” skorlaması ile ölçülecektir

## II. YÖNTEM

Bu projede kullanılan yöntem 5 aşamada anlatılacaktır.


**1.Aşama (Masaüstü Arayüzü Geliştirilmesi ve Graf Yapısının Oluşturulması):** Öncelikle arayüzde bizden beklenen isterler şunlardır:

- Kullanıcının doküman yükleyebileceği bir alan,
- Dokümanın graf halinde görüntüleneceği bir alan,
- Cümle benzerliği için threshold seçilebilecek bir araç,
- Cümle skorunun belirlenmesi için threshold seçilebilecek bir araç.
- Cümle benzerliği algoritmasına alternatif oluşturursanız bunun arayüzden seçilebilmesini sağlayan bir araç.

Kullanıcının programa doküman yükleyebilmesi için Java Swing kütüphanesinden JFileChooser class’ını kullandık. Bu sınıfın nesnesi bize bir buton yardımı ile bir doküman yükleme için dosya gezgini açar. Sonrasında seçtiğimiz dokümanı program içine alabilmemiz için file tipinde bir nesneye bu seçilmiş olan dokümanın bilgisayarda kayıtlı olduğu yerin yolunu veriyoruz.

Graf yapısının görüntülenebilmesi için “DrawingGraph.java” isimli class oluşturup MyFrame sınıfından gelen graf çizimi için gerekli olan verilerden sonra JUNG kütüphanesi kullanılarak çizim yapılacaktır.

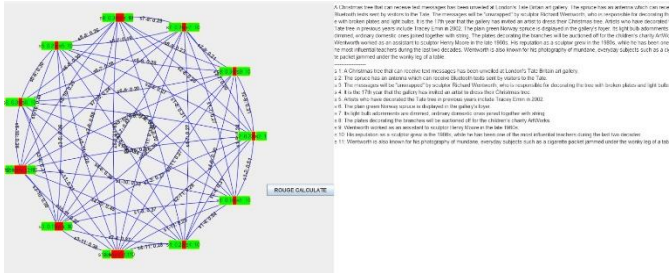
Cümle benzerliği ve skoru threshold seçimi için Main sınıfında 2 adet JTextField kullandık. Bu threshold değerleri 0 ile 1 arasında double değişken türünde olması gerektiğinden gerekli kontroller yapıldıktan sonra ilgili değişkenlere verilmiştir.



Threshold of Sentence Similarity :

Threshold of Sentence Score :

**Choose A File**



## 2.Aşama (Cümleler Arası Anlamsal İlişkinin Kurulması):

Cümlelere OpenNLP kütüphanesi kullanılarak aşağıdaki ön işleme adımları uygulanmıştır:

- **Tokenization:** Bir metnin küçük parçalara ayrılmasıdır.
- **Stemming:** Kelimelerin kökünün bulunması işlemidir.
- **Stop-word Elimination:** Bir metindeki gereksiz sözcükleri çıkarma işlemidir. Stop word'ler, genellikle yaygın olarak kullanılan, ancak metnin anlamını belirlemekte önemli bir rol oynamayan kelime ve ifadelerdir.
- **Punctuation:** Cümledeki noktalama işaretlerinin kaldırılmasıdır.

Cümleler arası anlamsal ilişkisinin kurulması için proje isterlerinden 2 yöntem vardır. Bunlar Word Embedding ve BERT algoritmalarıdır. Java için BERT algoritmasını kullanabilmek için yeterli sayıda model bulamadığımız için Word Embedding kullanmayı tercih ettik. Fakat kütüphane araştırması yaptığımızda yine yeterli bir kaynağa rastlamadık. Biz de Web API kullanan Jsoup kütüphanesi yardımı ile bir web adresi üzerinden GET edip Web Scraping yaparak anlamsal benzerlik sonucunu çıktık.

### 3.Aşama (Cümle Skoru Hesaplama Algoritmasının Geliştirilmesi):

Cümle skoru hesaplama sırasında aşağıdaki parametreleri oluşturduk :

1. Cümle özel isim kontrolü (P1)  
*Cümledeki özel isim sayısı / Cümlelerin uzunluğu*
2. Cümlede numerik veri olup olmadığının kontrolü (P2)  
*Cümledeki numerik veri sayısı / Cümlelerin uzunluğu*
3. Cümle benzerliği threshold'unu geçen node'ların bulunması (P3)  
*Thresholdu geçen nodeların bağlantı sayısı / Toplam bağlantı sayısı*
4. Cümlede başlıktaki kelimelerin olup olmadığının kontrolü (P4)  
*Cümledeki başlıkta geçen kelime sayısı / Cümlelerin uzunluğu*
5. Her kelimenin TF-IDF değerinin hesaplanması (P5).  
Buna göre dokümandaki toplam kelime sayısının yüzde 10'u 'tema kelimeler' olarak belirlenmelidir.

*Cümlelerin içinde geçen tema kelime sayısı / Cümlelerin uzunluğu*

#### 4.Aşama (Skorlara Göre Metin Özetleme Algoritmasının Geliştirilmesi):

Öncelikle bu aşamada, verilen dokumanın uygun bir özetini çıkarmak için dokumanın konusundan ve anlatmak istediği çizgiden sapma olmadan uygun cümleleri özete eklememiz gerekir. Bunun için kendimizce 3. aşamada hesaplamış olduğumuz 5 adet parametreleri (Cümle özel isim kontrolü, Cümlede numerik veri olup olmadığının kontrolü, Cümle benzerliği threshold'unu geçen node'ların bulunması, Cümlede başlıktaki kelimelerin olup olmadığının kontrolü, Her kelimenin TF-IDF değerinin hesaplanması) onların bir metin için olabilecekleri öneme göre ağırlık ataması yaptık. (P1 : 2, P2 : 1, P3 : 5, P4 : 3, P5 : 4) Böylece her bir cümlenin bulunduğu node'un cümle skoru hesaplanmış olur. Sonrasında Main.java sınıfında almış olduğumuz cümle skoru thresholdunu kullanarak dokumanın başından itibaren bu threshold'dan yüksek olup olmadığını kontrol ettik. Eğer öyleyse özetimize bu cümleleri ekledik.

Generated Summary	Reference Summary
<p>lots who have decorated the Tate tree in previous years include Tracey Minn in 2002. The plain green Norway spruce is displayed in the gallery's foyer. Its light bulb adornments are dimmed, ordinary domestic ones joined together with string. The plates decorating the branches will be auctioned off to the children's charity ArtWorks. Wentworth worked as an assistant to sculptor Henry Moore in the late 1960s. His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades. Wentworth is also known for his photography of mundane, everyday subjects such as a cigarette packet jammed under the wonky leg of a table.</p> <p>precision: 0.89 recall: 0.97 f1score: 0.98</p>	<p>The plain green Norway spruce is displayed in the gallery's foyer. Its light bulb adornments are dimmed, ordinary domestic ones joined together with string. The plates decorating the branches will be auctioned off to the children's charity ArtWorks. Wentworth worked as an assistant to sculptor Henry Moore in the late 1960s. His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades. Wentworth is also known for his photography of mundane, everyday subjects such as a cigarette packet jammed under the wonky leg of a table.</p> <p>Calculate...</p>

## 5.Aşama (Özetleme Başarısının ROUGE Skoru ile Hesaplanması):

Rouge-N algoritmasında n-grams değerini 1 aldık. Bu da tek bir kelime üzerinden 2 metnin Rouge skoru hesaplanmasını sağladık. Rouge skoru 3 parametreden oluşmaktadır. (Precision, Recall, F1Score)

- Precision : Eşleşen kelime sayısı / Oluşturulmuş özetteki kelime sayısı
- Recall : Eşleşen kelime sayısı / Referans özetteki kelime sayısı
- F1Score :  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

Böylelikle Rouge-1 skoru parametreleri hesaplanmış olur.

Oluşturulan Classlar :

- DrawingGraph.java
- Main.java
- MyFrame.java
- RougeCalculator.java
- TFIDFCalculator.java

Kullanılan programlar: Eclipse

## SÖZDE KOD

1-BAŞLA

2-Kullanıcıdan threshold değerlerini AL

3-EĞER kullanıcı butona tıklarsa GİT 4

4-EĞER alınan inputlar geçerli ise GİT 5 DEĞİLSE GİT 2

5-HESAPLA cümle skoru, cümle benzerliği vs. GİT 6

6- YAZDIR Graf ve özet GİT 7

7-EĞER Rouge butonuna tıklarsa GİT 8

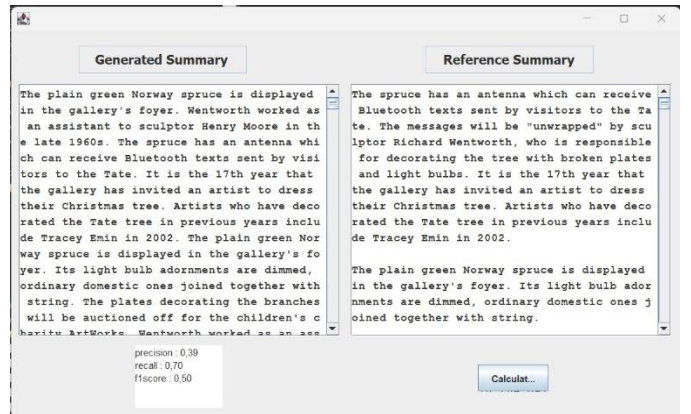
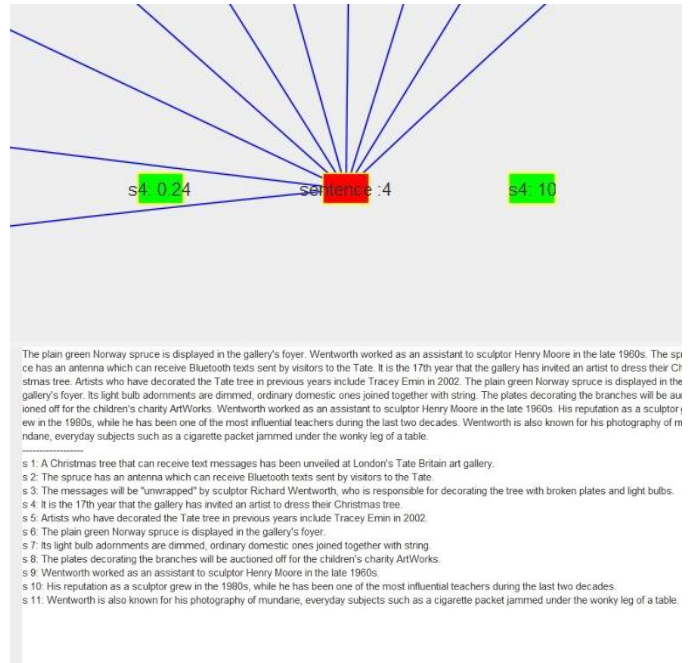
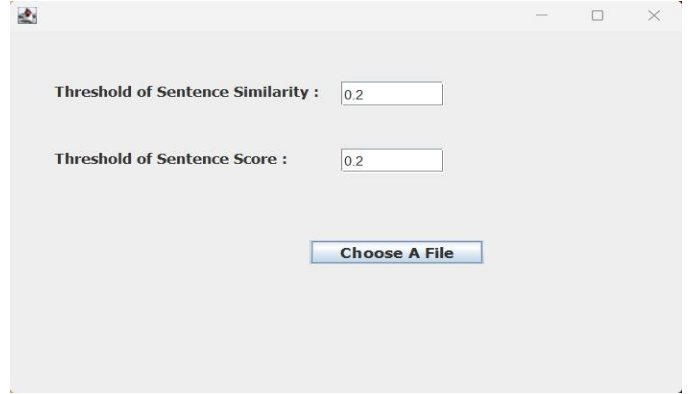
8- EĞER Calculate butonuna tıklarsa GİT 9

9- HESAPLA Rouge skorları GİT 10

10-YAZDIR Rouge skor

11-BİTİR

## III. DENEYSEL SONUÇLAR



#### IV. SONUÇ

Bu projede temel amaç cümleleri graf yapısına çevirip Cümle Seçerek Özetleme (Extractive Summarization) gerçekleştirmekti. Graf yapısına çevirerek cümlelerin metindeki anlamsal ilişkilerini görselleştirmek ve bu ilişkileri kullanarak önemli cümleleri belirlemeyi amaçlandı. Devamında bu cümlelerin hesaplanmış threshold bağlantı sayısı ve cümle benzerlik oranı ile birlikte graf gösterimi yapıldı.

Kitap, dergi makale vb. gibi düz yazı kaynaklarının veya bunların içindeki metinlerin programlama üzerinden doğal dil işleme yöntemleri kullanılarak nasıl gerçeğine en yakın ve anlamlı bir özetleme yapılacağını öğrendik. Bu işlemlerin yapılması için gerekli algoritma yapılan araştırmalar ışığında uygulamaya çalıştık.

#### KAYNAKLAR

- <https://youtu.be/A6sA9KItpwY> bro code
- <https://youtu.be/IHFIAYaNfdo> Alex Lee Java read text file easily
- <https://youtu.be/omNesNNSHTg> Maven Projesi
- [https://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html) part of speech
- <https://stackoverflow.com/questions/18590901/check-and-extract-a-number-from-a-string-in-java> --numeric var mı içinde
- <https://stackoverflow.com/questions/12806278/double-decimal-formatting-in-java> -- double formatlama
- <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html> -- stemming lemmatization
- <https://gist.github.com/sebleier/554280> stopwords list
- <https://www.techie-knowledge.co.in/2017/02/removing-stop-words-from-text-using-java.html> ---- hashset stopwords kaldırma
- <https://stackoverflow.com/questions/5520693/in-java-remove-empty-elements-from-a-list-of-strings>
- <https://huggingface.co/models> --- modeller bert ve word embeddin için
- <https://huggingface.co/bert-base-cased/tree/main> --- modeller bert ve word embeddin için
- <https://stackoverflow.com/questions/2037832/semantic-similarity-between-sentences> ----- benzerlik için
- <https://www.tutorialkart.com/opennlp/apache-opennlp-tutorial/> -- opennlp tutorial
- <https://www.tutorialspoint.com/opennlp/index.htm> --- opennlp tutorial
- <https://www.dailysmarty.com/posts/full-list-of-stopwords-as-a-ruby-array> -- stopwords için full
- <https://www.techiedelight.com/remove-punctuation-from-string-java/#:~:text=The%20standard%20solution%20to%20remove,expression%20that%20finds%20punctuation%20characters.> --- noktalama işaretleri kaldırmak için
- <https://jar-download.com/online-maven-download-tool.php> --- maven dependency to jar file
- <https://gist.github.com/guenod/d5add59b31114a3a3c66> --- tf-idf hesabı için
- <https://mdurmuss.github.io/tf-idf-nedir/> --- tf-idf bilgi
- <https://stackoverflow.com/questions/8115722/generating-unique-random-numbers-in-java> unique random numbers generate
- <http://swoogle.umbc.edu/SimService/index.html> --- semantic benzerlik için kullandım
- <https://github.com/tdebatty/java-string-similarity> ---cosinus benzerliği için kullandım
- <https://tilores.io/cosine-similarity-online-tool> --- online cosinus benzerliği için kullanıldı check için
- <https://www.tabnine.com/code/java/methods/opennlp.tools.stemmer.PorterStemmer/stem> --- stemmerin kullanımı opennlp
- <https://opennlp.apache.org/docs/2.2.0/apidocs/opennlp-tools/opennlp/tools/stemmer/snowball/SnowballStemmer.html> ---- snowball kullanımı
- <http://text-processing.com/demo/stem/> ---- stemmer için demo online web sitesi kontrol için kullandım
- <https://www.geeksforgeeks.org/how-to-convert-hashmap-to-arraylist-in-java/> ---- hashmap i arrayliste çevirmek için kullandım
- <https://stackoverflow.com/questions/6026813/converting-string-array-to-java-util-list> --- string to arraylist için kullandım
- rouge hesabı için chatcpt kullanıldı.
- <https://youtu.be/TMshhnrEXlg> rouge nedir
- <https://stackabuse.com/java-check-if-string-is-a-number/> ----- String numeric veri içeriyor mu kontrolü